NORTH-HOLLAND

Numerical Solution of the Lyapunov Equation by Approximate Power Iteration*

A. Scottedward Hodel† and Bruce Tenison‡
Department of Electrical Engineering
Auburn University
200 Broun Hall
Auburn, Alabama 36849

and

Kameshwar R. Poolla§
Department of Mechanical Engineering
University of California
Berkeley, California 94720

Submitted by Richard A. Brualdi

ABSTRACT

We present the approximate power iteration (API) algorithm for the computation of the dominant invariant subspace of the solution X of large-order Lyapunov equations $AX + XA^T + Q = 0$ without first computing the matrix X itself. The API algorithm is an iterative procedure that uses Krylov subspace bases in computing estimates of matrix-vector products Xv in a power iteration sequence. Application of the API algorithm requires that $A + A^T < 0$; numerical experiments indicate that, if the matrix X admits a good low-rank solution, then API provides an orthogonal basis of a subspace that closely approximates the dominant X-invariant subspace of corresponding dimension. Analytical convergence results are also presented.

*This work was supported in part by National Science Foundation contracts ECS 87-09265-EIA and ECS-9110083. A preliminary version of this paper was presented at the 1990 IEEE Conference on Decision and Control.

†E-mail: scotte@eng.auburn.edu
‡E-mail: rbtenis@ibm.com
§E-mail: poolla@jagger.berkeley.edu

LINEAR ALGEBRA AND ITS APPLICATIONS 236:205–230 (1996)

655 Avenue of the Americas, New York, NY 10010

0024-3795/96/$15.00
SSDI 0024-3795(94)00155-7

1. INTRODUCTION

The Lyapunov equation

$$AX + XA^T + Q = 0, \qquad A, Q \in \mathbb{R}^{n \times n}, \quad Q = Q^T \geqslant 0, \tag{1.1}$$

plays a significant role in numerous problems in control, communication systems theory, and power systems. Standard methods for the numerical solution of the Lyapunov equation [3, 7] make use of the real Schur decomposition $A = USU^T$, where U is an orthogonal matrix and S is quasi-upper-triangular. The Bartels-Stewart algorithm [3] is the method of choice for the solution of general small ($n \leqslant 100$), dense Lyapunov equations, while the Hammarling algorithm [7] may be preferred when a Cholesky factorization $Q = BB^T$ is available. Parallel implementation of these algorithms is considered in [21] and [13], respectively. Recent work addresses iterative methods for the numerical solution of *large* Lyapunov equations where the coefficient matrix A is sparse [13, 26, 27, 14, 15, 24].

In this paper we consider the problem of computing an estimate of the dominant *low-rank* invariant subspace of the exact solution X of the Lyapunov equation (1.1) *without* first computing the matrix X itself. Our motivation for this problem arises from numerical algorithms related to reduced order modeling and control [2, 20]. Our assumptions are that (1) the matrix A is stable, (2) the matrix Q is symmetric positive semidefinite, and (3) the structure of the matrix A admits efficient matrix-vector products Av, A^Tv with arbitrary vectors $v \in \mathbb{R}^n$. Under these assumptions it follows that the solution X of the Lyapunov equation is symmetric positive semidefinite [6]. Since X is only specified implicitly, our algorithm uses estimates of the matrix-vector product $y = Xv$ in order to iteratively compute an orthogonal matrix $V \in \mathbb{R}^{n \times k}$ whose span approximates the dominant invariant subspace of the solution X of the Lyapunov equation (1.1).

The remainder of this paper is organized as follows. In Section 2 we summarize issues relating to Krylov-subspace based algorithms and to the API algorithm. Following this, in Section 3 we present the API algorithm in detail and discuss algorithm convergence issues and error analysis. We then discuss in Section 4 numerical experiments that illustrate both algorithm effectiveness and algorithm requirements in order to obtain an acceptable subspace basis estimate V. Finally, in Section 5 we summarize our analysis and draw some conclusions.

2. KRYLOV SUBSPACES AND REDUCED-ORDER SOLUTION METHODS

Numerical solution of the Lyapunov equation (1.1) has been addressed for several decades [3, 7, 8, 13–15, 17, 19, 24, 26, 27]. Early algorithms employed a Kronecker product expansion that transformed the Lyapunov equation into a large, sparse linear system of n^2 equations with n^2 unknowns, requiring $O(n^6)$ flops for numerical solution. Practical solution procedures, beginning with the Bartels-Stewart algorithm [3], make direct use of the matrix-valued structure of the Lyapunov equation, which allows numerical solution in $O(n^3)$ flops. Iterative techniques, such as successive overrelaxation (SOR), alternating-direction implicit (ADI) [5, 26, 27], and Krylov-subspace based residual minimization schemes [14], are recommended for Lyapunov equations where the coefficient matrix A is large and sparse. SOR and ADI methods require information on the spectrum $\lambda(A)$ and provide *full-rank* estimates $\hat{X} \approx X$ at each iteration.

Solutions of large Lyapunov equations (1.1) frequently admit good low-rank approximations. Furthermore, we are frequently interested in computing only the dominant invariant subspace of X (or some related subspace) rather than the matrix X itself [20, 25]. We therefore propose to identify directly the dominant rank-k invariant subspace of the Lyapunov-equation solution $X \in \mathbb{R}^{n\times n}$, where $k \ll n$. Our algorithm makes extensive use of Krylov-subspace techniques. We therefore summarize their properties as follows.

DEFINITION 2.1. Given a matrix $A \in \mathbb{R}^{n\times n}$ and a vector $z \in \mathbb{R}^n$, the *Krylov subspace* $\mathrm{Kr}(A, z, k)$ is defined as $\mathrm{Kr}(A, z, k) = \mathrm{span}([z\ Az\ \cdots\ A^{k-1}z])$. Similarly, a *block Krylov subspace* $\mathrm{Kr}(A, Z, k)$ for matrices $Z \in \mathbb{R}^{n\times k_1}$ is defined as $\mathrm{Kr}(A, Z, k) = \mathrm{span}([Z\ AZ\ \cdots\ A^{k-1}Z])$.

DEFINITION 2.2. A *Krylov-subspace basis* V_k (or V when the context is clear) is an orthogonal matrix $V_k \in \mathbb{R}^{n\times k}$ such that $\mathrm{span}\, V_k = \mathrm{Kr}(A, z, k)$ or, for a block Krylov subspace, $\mathrm{span}\, V_k = \mathrm{Kr}(A, Z, k)$.

A Krylov-subspace basis V may be computed by the well-known Arnoldi algorithm; see, e.g., [23]. The Arnoldi algorithm sequentially computes an orthogonal matrix $V = [v_1\ \cdots\ v_k]$ such that $\mathrm{span}(v_1, \ldots, v_k) = \mathrm{Kr}(A, z, k)$. Alternatively, Householder reflections may be used in order to avoid the loss of orthogonality in V_k that often occurs in the Arnoldi process; see [6] for details. Krylov-subspace based methods for the numerical solution of the

Lyapunov equation make use of the Krylov-subspace basis V to provide *low-rank* updates $\hat{X} = V\Sigma V^T$ in order to compute the Lyapunov equation solution X; error bounds may often be computed *a priori* in terms of the spectrum $\lambda(A)$ in order to monitor algorithm progress. A generic Krylov-subspace based iterative algorithm for the numerical solution of the Lyapunov equation is presented below; the algorithm is adapted from [14].

ALGORITHM 2.1. [Krylov-subspace methods for iterative solution of the Lyapunov equation (1.1)].

Inputs: $A, Q \in \mathbb{R}^{n\times n}$.

Outputs: $X \in \mathbb{R}^{n\times n}$ satisfying Equation (1.1).

1. $R_0 = Q$; $i = 0$; $X_0 = 0$.
2. While $\|R_i\|$ is too large:
   (a) z_i = column of R_i with maximum norm.
   (b) Compute an orthogonal basis $V_i \in \mathbb{R}^{n\times k}$ of the Krylov subspace $\mathrm{Kr}(A, z_i, k)$.
   (c) Compute $\Sigma_i \in \mathbb{R}^{k\times k}$ by either a least-squares minimization [11, 14, 15]

$$\min_{\Sigma\in\mathbb{R}^{k\times k}} \|AV_i\Sigma V_i^T + V_i\Sigma V_i^T A^T + R_i\|_F \tag{2.1}$$

   or a reduced-order Lyapunov equation [12, 15, 24]

$$(V_i^T A V_i)\Sigma_i + \Sigma_i(V_i^T A^T V_i) + V_i^T R_i V_i = 0. \tag{2.2}$$

   (d) $\hat{X}_{i+1} = \hat{X}_i + V_i\Sigma_i V_i^T$; $i = i + 1$.
   (e) $R_i = -(A\hat{X}_i + \hat{X}_i A^T + Q)$.
3. End while.
4. $X = \hat{X}_i$.

REMARK 2.1. Algorithm 2.1 may be modified in step 2(a) to use block Krylov subspaces by selecting a matrix Z_i whose columns are drawn from the columns of R_i with maximum norm.

REMARK 2.2. Minimizations of the form (2.1) can be efficiently solved via a conjugate-gradient algorithm; see [11] or [14]. The reduced-order Lyapunov equation (2.2) requires only standard numerical techniques [3, 7].

REMARK 2.3. Saad [24] shows that the solution of a reduced-order Lyapunov equation (2.2) is equivalent to a Galerkin approximation of the integral form

$$X = \int_0^\infty e^{At} Q e^{A^T t}\, dt$$

of the Lyapunov equation solution X.

Practical error bounds [14, 24] are provided for both the least-squares and the reduced-order Lyapunov-equation methods. Hu and Reichel [14] present error bounds for their algorithms in terms of the Ritz values of the matrix $V^T AV$ at each iteration; Saad [24] presents error bounds for Galerkin approximation methods in terms of the following theorem.

THEOREM 2.1 [24]. *Let $A \in \mathbb{R}^{n\times n}$ be an arbitrary matrix, and let $v \in \mathbb{R}^n$ be an arbitrary unit vector. Let V_j be a Krylov subspace basis of* $\mathrm{Kr}(A, v, j)$. *Define $a = \|A\|_2$ and $\Theta_j = V_j^T A V_j$. Then*

$$\|e^{At} v - V_j e^{\Theta t} e_1\|_2 \leq \frac{2(at)^j e^{at}}{j!}.$$

The error bounds discussed above should not be interpreted to be saying that reduced-order Lyapunov equations may be used indiscriminately in algorithms for low-rank approximate solution of Lyapunov equations. Consider the following lemma.

LEMMA 2.1 [10, 12]. *Let $A \in \mathbb{R}^{n\times n}$ be a real, stable matrix. If $A + A^T < 0$, then the reduced-order Lyapunov equation (2.2) admits a unique solution Σ_V for all orthogonal $V \in \mathbb{R}^{n\times k}$. Conversely, if $A + A^T$ is not negative definite, then (1) there exists an orthogonal matrix V_1 such that the associated reduced-order Lyapunov equation does not admit a unique solution, and (2) there exists an orthogonal matrix V_2 such that $V_2^T A V_2$ is strictly unstable (all eigenvalues are in the closed right half plane).*

REMARK 2.4. The above lemma indicates that if $A + A^T$ is not negative definite, then it is possible to obtain reduced-order Lyapunov equations that are arbitrarily ill conditioned. As shall be seen in Sections 3 and 4, this negative-definiteness condition bears considerable weight when applying the API algorithm presented in Section 3.

It is readily shown that, given an orthogonal basis V of an X-invariant subspace, the estimate $X = V\Sigma V^T$ obtained through a reduced-order Lyapunov equation exactly recovers $\Sigma = V^T X V$; that is, if a basis V of the dominant eigenspace of X can be computed, then the two-norm optimal rank-k estimate $\hat{X}_k$ of X may be obtained through a reduced-order Lyapunov equation. While no corresponding invariant-subspace based result exists for residual-minimization scheme (2.1), the following bounds may be computed for Lyapunov equations with $A + A^T < 0$.

LEMMA 2.2 [12]. *Let X be the solution of the Lyapunov equation* (1.1), *and let $\hat{X} = V\Sigma V^T$ be a low-rank estimate of X. Define the residual $R(\hat{X}) = A\hat{X} + \hat{X}A^T + Q$, and the* log norm $\mu(A)$ *of A as $\mu(A) = \max \lambda(A^T + A)/2$. If $\mu(A) < 0$, then*

$$\frac{\|R(\hat{X})\|_2}{2\|A\|_2} \leq \|X - \hat{X}\|_2 \leq \frac{1}{-2\mu(A)}\|R(\hat{X})\|_2.$$

3. APPROXIMATE POWER ITERATION

We present in this section the theoretical development and convergence analysis of the approximate power iteration (API) algorithm. Consider the Lyapunov equation (1.1). We shall henceforth assume that the matrix A is stable, that $Q = Q^T \geq 0$, and that (A, Q) is a controllable pair. Under these conditions, the solution X of the Lyapunov equation is positive semidefinite. If the matrix X were specified explicitly, then we could estimate the dominant eigenspace of X using either power iteration or Lanczos methods [6, 22]. These methods require the ability to compute the product $y = Xv$ of the matrix X and an arbitrary vector $v \in \mathbb{R}^n$. Unfortunately, since X is specified only implicitly, as the solution of the Lyapunov equation (1.1), we cannot directly apply these methods to our problem.

While existing Krylov-subspace based methods permit the numerical computation of a *fixed-rank* estimate $\hat{X}$ of the solution X of the Lyapunov equation (1.1), they do not attempt to directly identify the dominant rank-k-invariant subspace of the solution X of the Lyapunov equation (1.1). More precisely, if one is dissatisfied with the computed estimate $\hat{X}$ of X, it is necessary to increase the rank of the estimated solution via either (1) increasing the dimension of the underlying Krylov subspace or (2) using the Krylov-subspace approach as a basis for an iterative method to compute the *exact* solution X of the Lyapunov equation (1.1).

3.1. Algorithm Description

The API algorithm is a modification of the generic Krylov subspace Algorithm 2.1 for the numerical solution of the Lyapunov equation (1.1). API employs a sequence of *block* Krylov subspace bases V_i with one or more free parameter vectors in the initializing matrices Z_i. The free parameter vectors are selected in an attempt to identify the dominant subspace of the Lyapunov-equation solution X without first computing X itself. The derivation of the API algorithm requires the assumption that $A + A^T$ is negative definite (i.e., $\|e^{At}\|_2$ is monotonically decreasing), and may be developed as follows. Let $V \in \mathbb{R}^{n\times k}$, $k \ll n$, be an arbitrary orthogonal matrix ($V^TV = I_k$), and consider the product $Y = XV$. Postmultiplication of the Lyapunov equation (1.1) by the matrix V yields

$$AY + Y\Theta + QV + XE = 0, \tag{3.1}$$

where $E \triangleq (I - VV^T)A^TV$, so that $E^TV = 0$. Notice that if the error term $\|XE\|$ is small, then we may attempt to compute an estimate $\hat{Y}$ of Y by solving the Sylvester equation

$$A\hat{Y} + \hat{Y}\Theta + QV = 0. \tag{3.2}$$

[Observe that $A + A^T < 0$ guarantees the existence of a unique solution of Equation (3.2) for all orthogonal $V \in \mathbb{R}^{n\times k}$.] In the event that the error $\|E\|$ in Equation (3.1) is unacceptable large ($\|XE\|$ is not available for measure), then an orthogonal basis W of the block Krylov subspace $\mathrm{Kr}(A^T, V, l)$ may be constructed for an appropriate integer l; the desired estimate $\hat{Y}$ may be computed by first solving the Sylvester equation

$$A\hat{Y}_1 + \hat{Y}_1W^TA^TW + QW = 0 \tag{3.3}$$

for $\hat{Y}_1$ and then extracting

$$\hat{Y} = \hat{Y}_1\begin{bmatrix} I_k \\ 0 \end{bmatrix},$$

i.e., $\hat{Y}$ is the first k columns of $\hat{Y}_1$.

REMARK 3.1. Higham [9] has shown that the Sylvester equation

$$AX + XB + C = 0, \qquad A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{m\times m}, \quad C \in \mathbb{R}^{n\times m},$$

is *not* backward stable; that is, for general matrices A, it is inappropriate to assume that the matrix $\hat{Y}_0$ obtained from Equation (3.2) is a good approxima-

tion of the product $Y_0 = XV_0$, regardless of how small $\|E\|$ may be. However, in the case $A + A^T < 0$, the analysis of Hewer and Kenney [8] may be adapted in order to quantitatively express an error bound that justifies the above approximation; the result is similarly justified from Lemma 2.2.

Further insights into the properties of Equation 3.2 are seen in the following lemma.

LEMMA 3.1. *Let* $A, Q \in \mathbb{R}^{n\times n}$ *with* $A + A^T < 0$, $Q = BB^T \geqslant 0$, *and* (A, Q) *controllable, and let* X *satisfy the corresponding Lyapunov equation* (1.1). *Let* $V \in \mathbb{R}^{n\times k}$ *be an arbitrary orthogonal matrix and define* $Y = XV$. *Let* $\hat{Y}$ *satisfy the Sylvester equation* (3.2). *Then the error* $E \triangleq Y - \hat{Y}$ *satisfies*

$$E \triangleq Y - \hat{Y} = \int_0^\infty e^{At}B\left(B^TVe^{\Theta t} - B^Te^{A^Tt}V\right)dt,$$

where $\Theta = V^TA^TV$.

Lemma 3.1 states that the error $Y - \hat{Y}$ is driven by he mismatch between $B^TVe^{\Theta t}$ and $B^Te^{A^Tt}V$. When V is constructed as the basis of a Krylov subspace, Saad [24] provides error bounds on the difference $e^{A^Tt}V - Ve^{\Theta t}$ that indicate the error will tend to decrease as the dimension k is increased. However, in the case that span $V \cap$ span $B = 0$ (the subspaces are orthogonal to one another), this behavior is of no benefit, since in that case

$$E \triangleq Y - \hat{Y} = \int_0^\infty e^{At}B\left(B^TVe^{\Theta t} - B^Te^{A^Tt}V\right)dt = -\hat{Y};$$

that is, the computed estimate has a relative error of 100%. An API algorithm, based on the above observations, is as follows.

ALGORITHM 3.1 (Approximate power iteration).

Inputs: $A, Q \in \mathbb{R}^{n\times n}$, $Q = BB^T \geqslant 0$, and integers $k, k_{\max}$.

Outputs: $\hat{V}$, an orthogonal basis of an estimate of the dominant eigenspace of the solution X of the Lyapunov equation (1.1).

1. Select $V_0 \in \mathbb{R}^{n\times k}$, orthogonal, via (e.g.) Algorithm 5.2 in [14] and a Krylov-subspace iteration.
2. $V_{-1} = 0$, $i = 0$.
3. While $\|V_i - V_{i-1}\|$ is large:
   (a) Compute an orthogonal basis $\bar{V}_i \in \mathbb{R}^{n\times k_{\max}}$ of the block Krylov subspace $\mathrm{Kr}(A, [V_i \;\; B], j)$, where j is the smallest integer such that $k_{\max} \leqslant jk$.

(b) Solve for $\hat{Y}_i$: $A\hat{Y}_i + \hat{Y}_i(\overline{V}_i^T A^T \overline{V}_i) + Q\overline{V} = 0$.

(c) Compute $V_{i+1} \in \mathbb{R}^{n\times k}$, an orthogonal basis of $\hat{Y}_i \begin{bmatrix} I_k \\ 0 \end{bmatrix}$; $i = i + 1$.

4. End while.
5. $V = V_i$.

REMARK 3.2. The loop condition in step 3 in the API algorithm 3.1 provides a measure of the separation between the subspaces span V_i and spanV_{i-1}, and is efficiently implemented as

$$\| V_{i-1} - V_i(V_i^T V_{i-1}) \|.$$

REMARK 3.3. Observe that step 3(a) computes a block Krylov subspace initialized with the matrix $[V_i \ B]$; in this sense, the API algorithm can be regarded as a Krylov-subspace iteration with a free parameter V_i that is adjusted in order to estimate the dominant rank-k subspace of the Lyapunov-equation solution X.

REMARK 3.4. Each iteration of the API algorithm requires the numerical solution of a Sylvester equation (3.2) where the matrix $A \in \mathbb{R}^{n\times n}$ is large and sparse and the matrix $\Theta \in \mathbb{R}^{l\times l}$ is small ($l \ll n$) and *dense*. If Θ is too large for standard linear solution techniques, then Equation (3.2) can be solved iteratively either by an adaptation of the parallel solution algorithm [13] or by using an adaptation of the ADI algorithm [18]. Alternatively, step 3(b) of the API algorithm 3.1 may be modified to solve either a least-squares minimization or a reduced-order Lyapunov equation as in step 2(c) of the generic Krylov-subspace algorithm 2.1. These approaches both reduce the computational burden in each step and preserve the underlying symmetry in the problem. These methods are compared in Section 4.

3.2. *Convergence and Error Analysis of* API

If $A + A^T < 0$, then all of the eigenvalues of A and $\Theta = V^T A^T V$ lie in the open left half plane, and an analysis similar to that of Lemma 2.2 may be used to show that

$$\| Y - \hat{Y} \|_2 \leq \frac{1}{2\mu(A)} \| XE \|_2 .$$

In the case where span V closely approximates the desired dominant eigenspace of the solution X of the Lyapunov equation (1.1), then $E^T V = 0$ implies that $\|XE\|_2$ is "small" relative to the magnitudes of the largest eigenvalues of X. Regrettably, as shall be seen in Section 4, a more precise analysis of the general API algorithm ($k_{\max} > 1$) is not likely to be forthcoming. While numerical experiments demonstrate the utility of the algorithm when the passivity assumption $A + A^T < 0$ holds, these same results also indicate that any formal error analysis is likely to be highly conservative in nature. Hence, our convergence analysis of the API algorithm serves as heuristic justification, and is presented solely for the case $k_{\max} = 1$.

Theorem 2.1 may be used to show that, when the API algorithm is used with $k_{\max} = 1$,

$$\|\hat{Y}_i - XV_i\|_2 \leqslant \int_0^\infty \|e^{At}Q\|_2 \frac{2(ta)^k e^{at}}{(k-1)!}\, dt \triangleq \delta,$$

where $a = \|A\|_2$. Thus, this crude analysis indicates that a stable fixed point of the algorithm satisfies

$$\|V_\infty - U_k U_k^T V_\infty\|_2 \leqslant \delta \sum_{i=0}^{\infty} \left(\frac{\lambda_{k+1}}{\lambda_k}\right)^i = \frac{\delta}{1 - \lambda_{k+1}/\lambda_k},$$

where $V_\infty = \lim_{i\to\infty} V_i$, U_k is the matrix of the k dominant (orthogonal) eigenvectors of X, and $\lambda_1 \geqslant \cdots \geqslant \lambda_n$ are the eigenvalues of X.

When we restrict our attention to Krylov subspaces $\mathrm{Kr}(A^T, v, j)$ of dimension 1, then the solution of the Sylvester equation (3.2) is equivalent to computing

$$\hat{y} = -[A + \theta(v)I]^{-1}Qv, \qquad \theta(v) = v^T A^T v, \tag{3.4}$$

where $\theta \in \mathbb{R}$ is a scalar and $\hat{y} \in \mathbb{R}^n$ is a vector; that is, the vector $\hat{y}$ is obtained as the solution of a large, sparse system of linear equations. Essentially, this algorithm may be viewed as an iterative application of the mapping

$$\phi : \mathbb{R}^n \to \mathbb{R}^n : v \to \frac{-[A + \theta(v)I]^{-1}Qv}{\left\|[A + \theta(v)I]^{-1}Qv\right\|_2}, \tag{3.5}$$

where $\theta(v) = v^T A^T v$. (Note that Equation 3.5 *omits* the matrix B from the initialization step 3(a) in Algorithm 3.1.) It is easily shown that a vector v is a fixed point of the rank-1 API algorithm if v is a generalized eigenvector of

$$Qv + \lambda[A + \theta(v)I]v = 0.$$

Brouwer's fixed-point theorem [1] may be applied as in [10] to demonstrate the existence of a fixed point of the rank-1 API algorithm. Our numerical experiments suggest that under certain conditions there is at least one stable fixed point v and that, if the dominant eigenvalues λ_1 and λ_2 of X are well separated (i.e., $\lambda_1 \gg \lambda_2$), then there is a stable fixed point v "near" the dominant eigenvector u_1 of X. While we are unable to establish that this fixed point v is stable, we are able to establish the existence of an attractive domain around u_1 containing this fixed point v.

We first consider a general expression for the error $Xv - \hat{y}$ associated with the linear system of equations (3.4).

THEOREM 3.1. *Consider the solution X of the Lyapunov equation* (1.1) *with $A + A^T < 0$. Let $v \in \mathbb{R}^n$, and define the vector $\hat{y} \triangleq -[A + \theta(v)I]^{-1}Qv$. Then*

$$\|y - \hat{y}\|_2 \triangleq \|Xv - \hat{y}\|_2 \leqslant \kappa(A)\big(\lambda_1\|u_1 - v\|_2^2 + \lambda_2\big).$$

Here λ_1 and λ_2 are the two dominant eigenvalues of X, u_1 is the dominant eigenvector of X, and $\kappa(A) = \|A\|_2\|A^{-1}\|_2$ is the condition number of A.

Theorem 3.1 may be used to establish the following convergence result.

THEOREM 3.2. *Consider the solution X of the Lyapunov equation* (1.1). *Let u_1 be the dominant eigenvector of X, let $\lambda_1 \geqslant \lambda_2$ be the dominant eigenvalues of X, and let $\kappa = \kappa(A) = \|A\|_2\|A^{-1}\|_2$ be the condition number of A. Let $v_0 \in \mathbb{R}^n$ be a unit vector, and consider the single-vector API algorithm defined iteratively by $v_{i+1} = \phi(v_i)$.*

Suppose

$$\frac{\lambda_2}{\lambda_1} < \frac{1}{16(\kappa + 1)^2}. \tag{3.6}$$

Define the real numbers $R > r > 0$ by

$$R = \frac{1}{4(\kappa + 1)}\left(1 + \sqrt{1 - 16(\kappa + 1)^2 \frac{\lambda_2}{\lambda_1}}\right)$$

$$r = \frac{1}{4(\kappa + 1)}\left(1 - \sqrt{1 - 16(\kappa + 1)^2 \frac{\lambda_2}{\lambda_1}}\right)$$

Then:

(1) *If* $\|v_i - u_1\|_2 \leqslant R$, *then* $\|v_{i+1} - u_1\|_2 \leqslant R$.
(2) *If* v_0 *is chosen such that* $\|v_0 - u_1\|_2 < R$, *then*

$$\limsup_{k \to \infty} \|v_k - u_1\|_2 \leqslant r. \tag{3.7}$$

(3) *There exists a fixed point* v_∞ (*not necessarily attractive*) *for the API algorithm with* $\|v_\infty - u_1\|_2 \leqslant r$.

Proof. The proof involves straightforward, but tedious algebra, and is omitted here for brevity. Details may be found in [10]. ■

REMARK 3.5. Observe that if the solution X of the Lyapunov equation (1.1) has rank 1, then $\lambda_2 = 0$ and this makes $r = 0$. Therefore, in this case, the API algorithm has a stable fixed point at the dominant eigenvector u_1 of X.

4. NUMERICAL EXPERIMENTS

We now present the results of numerical experiments in which the API algorithm is applied to several test systems. All experiments are performed using Matlab on a Sun Sparc10 computer. We first present in Section 4.1 our experimental design, from which results are given in Section 4.2.

REMARK 4.6. Our numerical experiments use the (block) Arnoldi method with full reorthogonalization by Householder reflections.

4.1. Experimental Design

The API algorithm 3.1 is tested as follows, with several stable example systems. In each system, the solution X of the associated Lyapunov equation

is computed by the Bartels-Stewart algorithm [3]. Four performance criteria are measured at each iteration:

(1) *Power iteration error*: The error $E = \|\hat{Y}_i - XV_i\|_2$. This error quantifies the quality of the estimates $\hat{Y}$ provided by Equation (3.2).

(2) *Power iteration error with phase correction*: We found that the API algorithm 3.1 occasionally computes vectors that are of opposite sign to the desired vectors XV_i, that is, while the power iteration error can be significant, the API algorithm nevertheless computes an acceptable subspace basis estimate $\hat{Y}_i$ of the space span(XV_i). For the purposes of illustration, the columns of Y_i are postmultiplied by $D_i = \text{diag}(\pm 1, \ldots, \pm 1)$ with the signs selected in order to minimize the error $\|\hat{Y}_i D_i - XV_i\|_2$. It is this error that is most significant in the performance of the API algorithm.

(3) *Reduced-order Lyapunov equation error*: As discussed in Section 2, if the Krylov subspace basis V spans an X-invariant subspace, then $\Sigma = V^T XV$ is the solution of the reduced-order Lyapunov equation

$$V^T AV\ \Sigma + \Sigma V^T A^T V + V^T QV = 0.$$

The associated error presented in the error analysis is

$$E_r = \|\Sigma - V^T XV\|_2.$$

This error provides a measure of the quality of the eigenvalue estimates provided by the reduced-order Lyapunov equation associated with iteration of the API algorithm 3.1.

(4) *Subspace identification error*: The distance between the subspace span V_i and the desired dominant subspace of the solution X of the Lyapunov equation (1.1) is measured as $E = \|V_i - U_k U_k^T V_i\|_2$, where U_k is the matrix of the dominant k eigenvectors of X.

The algorithm is tested using Sylvester equations (3.1), reduced-order Lyapunov equations (2.2), and least-squares minimizations (2.1) for each API iteration. The least-squares minimization routine is expected to perform poorly, since the routine "looks through the wrong end of a telescope," i.e., the algorithm attempts to minimize only the residual $\|R\|_F$, and does not examine the solution $\|X\|_F$. The example systems used in our experiments are as follows.

EXAMPLE 4.1. The first example system is based on a finite-difference discretization of a one-dimensional heat-flow example:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2},$$

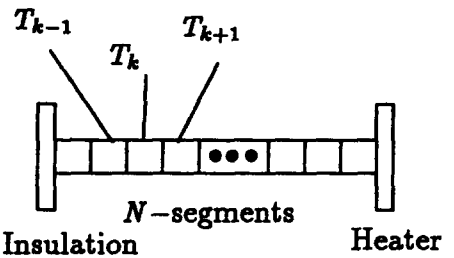


FIG. 1. One-dimensional heat-flow example.

where T is the temperature of the rod at position x and time t. The physical one-dimensional heat-flow system is shown in Figure 1. The resulting dynamical system is of the form

$$\dot{x} = Ax + Bu, \tag{4.1}$$

where $A \in \mathbb{R}^{250\times 250}$ is a tridiagonal matrix with diagonal entries $(-1/h, -2/h, -2/h, \ldots, -2/h)$ and $1/h$ on the off-diagonals, $B = [0 \cdots 0\ 1/h]^T$, and h is defined to be the step size between segments.

EXAMPLE 4.2. Our second example system is based on a two-dimensional extension of the one-dimensional heat-flow problem presented in Example 4.1. The partial differential equations are of the same form, and the system is discretized in a similar manner. The physical system of this example is shown in Figure 2. The states of the dynamical system (4.8), x_i, are the temperatures of the elements of the grid take in row-major order. The resulting matrices for the dynamical system are order n^2, but have at most five nonzero entries per row.

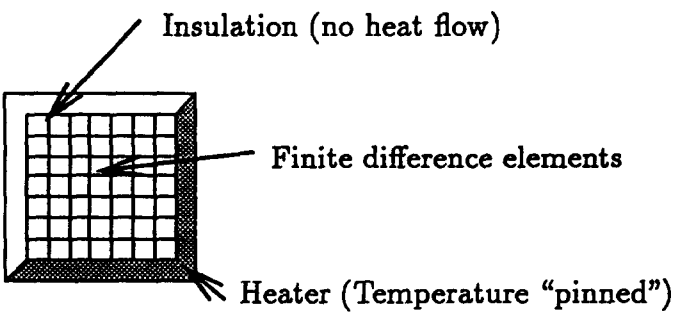


FIG. 2. Two-dimensional heat-flow example.

REMARK 4.7. Examples 4.1 and 4.2 both satisfy the passivity condition $A + A^T < 0$; we thus expect the API algorithm to perform well in these examples.

EXAMPLE 4.3. The third example system is based on the ACES example presented in [4]. This flexible test structure has 9 inputs, 10 outputs, and 86 states, each of which is highly oscillatory and lightly damped. Code is available to generate the model of this system (the system dynamic model is too large to present in this paper).

EXAMPLE 4.4. The next example consists of the cruciform example presented in [16], with rigid-body modes removed. This system is another flexible test structure which has modes that are highly oscillatory and lightly damped. This model has 34 states, 3 inputs, and 12 outputs. The system matrices are not included here; however, code is available to generate this model.

EXAMPLE 4.5. The fifth example system is based on an oscillating-spring model which satisfies partial differential equations of the form

$$M\frac{\partial^2 P}{\partial t^2} = K\frac{\partial^2 P}{\partial x^2} + D\frac{\partial P}{\partial t},$$

where P is the vertical position of the spring at horizontal position x at time t. The physical model of this system (order $2n$) is given in Figure 3. The system dynamics are given in Equation (4.8), with

$$A = \begin{bmatrix} 0 & I_n \\ A_{21} & -dI_n \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1/h^2 \end{bmatrix},$$

where A_{21} is a tridiagonal matrix with diagonal entries $(-k/h^2, -2k/h^2, -2k/h^2, \ldots, -2k/h^2)$ and k/h^2 on the off-diagonals. The variables k, d, and h are the spring constant between mass segments, the

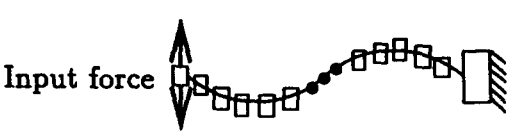


FIG. 3. Oscillating-spring example.

damping factor of each segment, and the step size between segments, respectively.

REMARK 4.8. Observe that Examples 4.3–4.5 do *not* satisfy the condition $A + A^T < 0$; thus, we expect the API algorithm to perform poorly on these examples.

4.2. Results

We now present the results obtained from the API iteration process on the above systems. First, results from the application of the algorithm to the one-dimensional heat diffusion problem (Example 4.1) with 250 states are presented in Figures 4–6. The dominant eigenvector of the solution to the Lyapunov equation (1.1) is estimated by the API routine ($k = 1$), and the maximum dimension of the block Krylov subspace allowed is 6, 8, 12, and 32 for four separate runs. Power iteration error plots are presented in Figure 4. The plots of power iteration error with phase correction for this example system are identical to those in Figure 4, and are omitted. Reduced-order Lyapunov-equation error plots are shown in Figure 5, and subspace identification error plots are shown in Figure 6.

REMARK 4.9. Figures 4–6 show that all error measures decrease quickly (within six iterations). However, after a number of iterations in the upper two plots of each figure, there is a "spike" in the estimation errors. Several comments are in order:

(1) This behavior was observed eventually in all API runs; this property of the algorithm is the key limiting factor in a theoretical convergence analysis of the API algorithm.

(2) The error spike can be delayed by increasing the dimension $k_{\max}$ of the block Krylov subspace used in the API algorithm.

(3) Since convergence and divergence of the computed subspaces span V_i can be easily detected on line, a practical API algorithm can be used that monitors the change in subspace $\|V_i - V_{i-1}V_{i-1}^T V_i\|$ and either (1) stops the iteration or (2) increases the dimension $k_{\max}$ of the API iteration when a large change in subspace is detected. Thus, the API algorithm, while not amenable to a formal convergence analysis, can still be applied intelligently and fruitfully with proper monitoring code within the iteration.

REMARK 4.10. A modified version of the API algorithm that uses reduced-order Lyapunov equations (2.2) in step 3(b) of the API algorithm 3.1 was tested on this example. As expected, computed errors are slightly

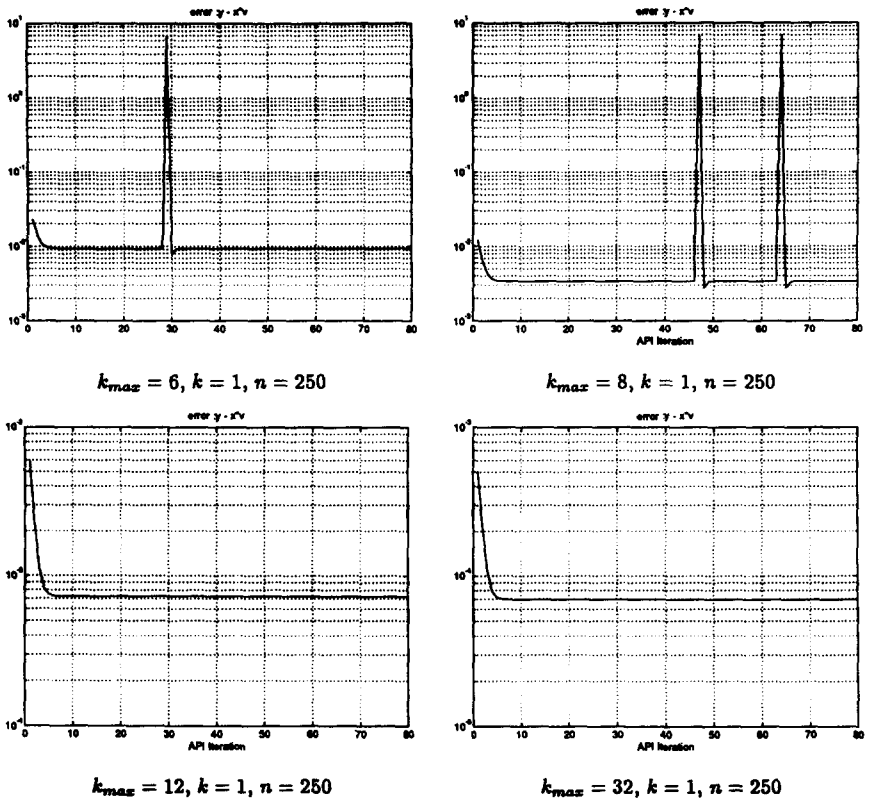


FIG. 4. API power iteration with 250th-order one-dimensional heat-flow example.

degraded from the standard API algorithm results; however, the algorithm behavior was essentially the same as for standard API. This mild loss of algorithm performance can thus be traded against the large gains in computational speed per iteration that are provided by using reduced-order Lyapunov equations instead of the standard API algorithm.

REMARK 4.11. The API algorithm was tested on this example, utilizing a least-squares minimization (2.1) in step 3(b) of Algorithm 3.1. The convergence behavior of this modified algorithm, using the same system as in Example 4.1 with 250 states, and estimating the rank-5 dominant subspace of the Lyapunov solution, is shown in Figure 7. As shown, the computed

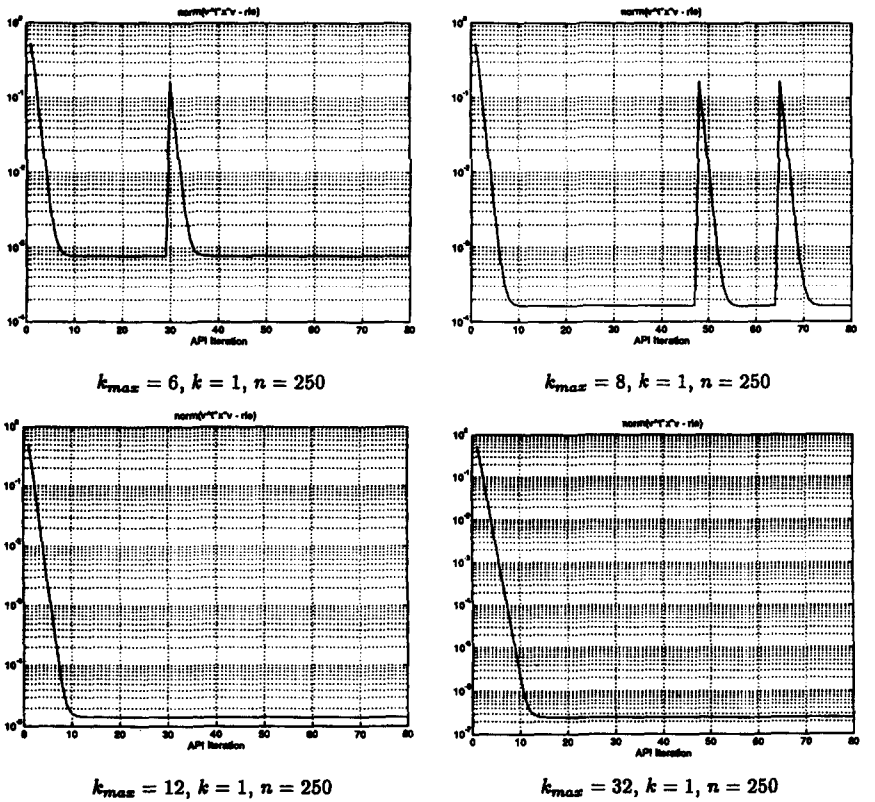
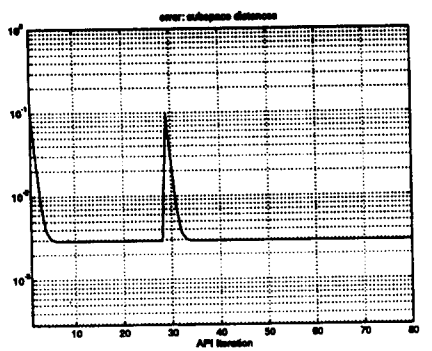


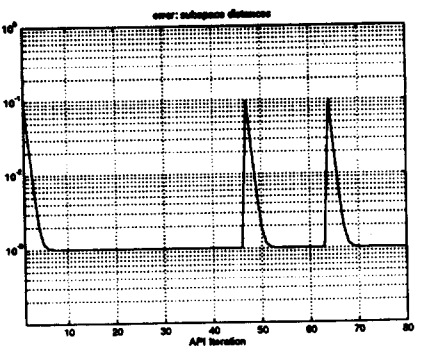
FIG. 5. API reduced-order Lyapunov-equation error with 250th-order one-dimensional heat-flow example.

performance measures display convergence behavior that is significantly degraded from that of the reduced-order Lyapunov equation procedure (Remark 4.10). This behavior is to be expected, since the least-squares approximation method, as discussed earlier (see Lemma 2.2 and discussion), does not exhibit the X-invariant subspace property.

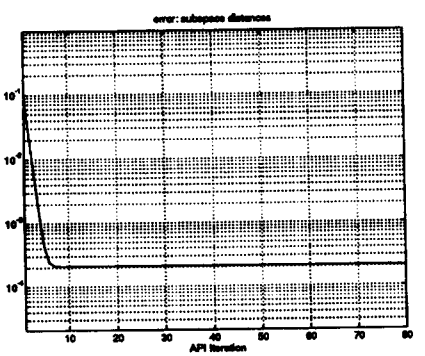
REMARK 4.12. The API algorithm was applied to a two-dimensional heat-flow example with a 15×15 grid (225 states). The results are essentially identical to those in the one-dimensional heat-flow example, and are omitted for brevity.



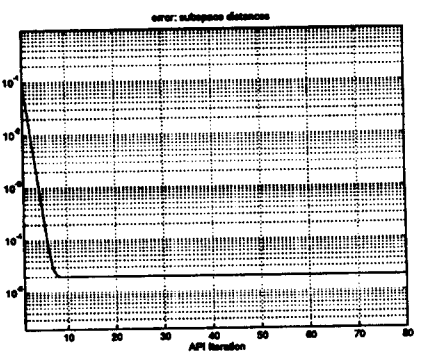
$k_{max} = 6, k = 1, n = 250$



$k_{max} = 8, k = 1, n = 250$



$k_{max} = 12, k = 1, n = 250$



$k_{max} = 32, k = 1, n = 250$

FIG. 6. API subspace identification error with 250th-order one-dimensional heat-flow example.

The API algorithm was also applied to Examples 4.3–4.5. These systems differ significantly from the first two in that they are based upon a wave-equation rather than a diffusion-equation model. As expected, the performance on these systems was poor, since the required condition $A + A^T < 0$ was violated. All the algorithm results for these systems were quite similar, and so only the results from Example 4.3 (the ACES structure) are presented here.

The API routine was tested on the ACES structure with $k_{\max} = 36$ and $k = 9$ [attempting to estimate the dominant rank-9-invariant subspace of the solution X of the Lyapunov equation (1.1)]. The four corresponding error plots are shown in Figure 8.

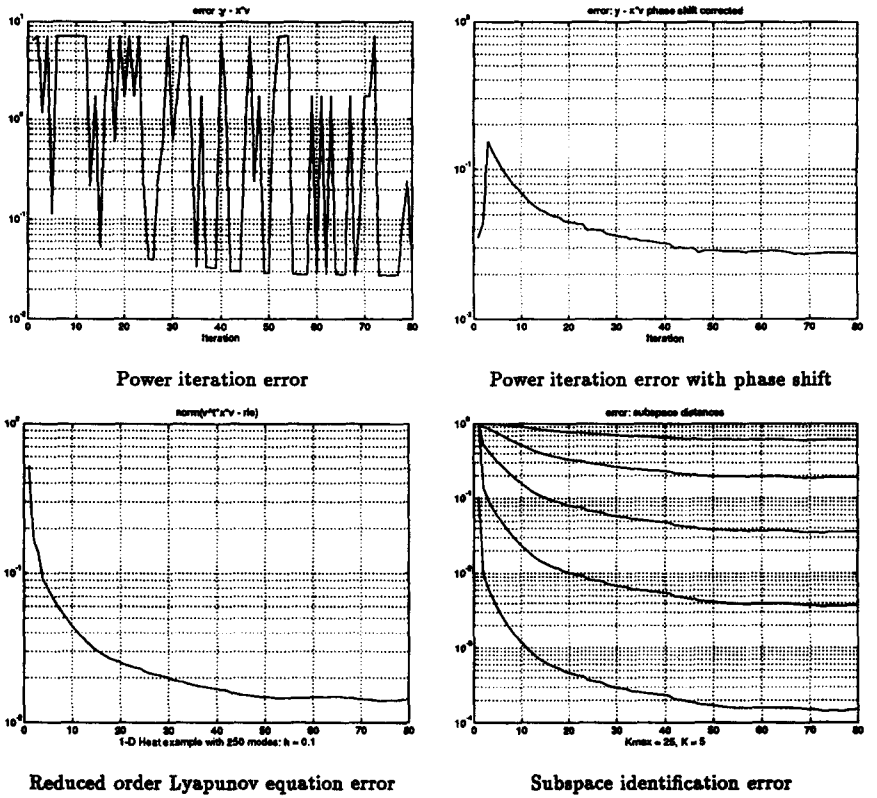


FIG. 7. One-dimensional heat-flow example, least-squares approximation.

REMARK 4.13. The API algorithm does occasionally provide good estimates of the dominant subspace of the Lyapunov equation solution X, but due to a lack of convergent behavior, it is impossible to determine in a straightforward fashion when such estimates may be obtained.

In order to apply the API algorithm to systems such as those in Examples 4.3–4.5, it is necessary to determine a preconditioning procedure (coordinate transformation) that (1) preserves the sparsity structure of the matrix A, and (2) yields a new dynamic system such that $A + A^T < 0$. As an extreme example, the system of Example 4.3 was diagonalized via a Jordan decomposition in order to examine the resulting algorithm behavior. (It must be emphasized that the Jordan decomposition cannot be regarded as a practical API preconditioner; this example is provided merely to establish the potential

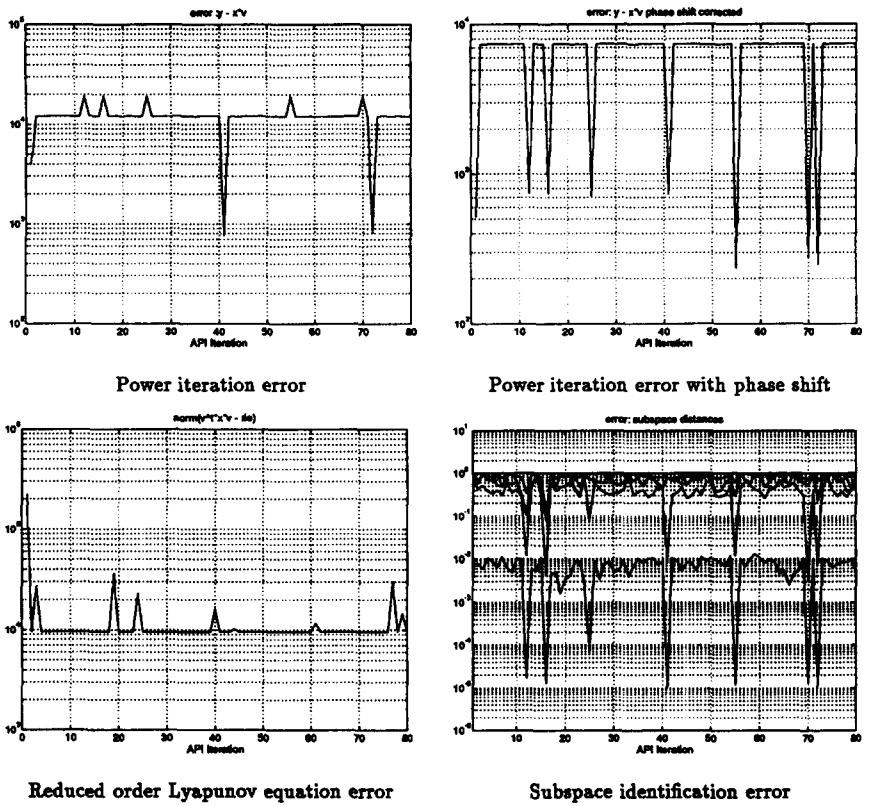


FIG. 8. API error: ACES example.

utility of such a preconditioning procedure.) Results of the API algorithm applied to the transformed model are shown in Figure 9. These results are certainly improved (reasonable estimates of the dominant eigenvector are occasionally available), but detection of convergence remains an open problem.

REMARK 4.14. The API algorithm was tested on the ACES example using the least-squares minimization procedure outlined in Remark 4.11. Figures 10 and 11 display the convergence-criterion plots of the algorithm utilizing both the "raw" and the Jordan block form of the ACES system, respectively. As above, the Jordan form exhibits better results than the "raw" form of the system, but still exhibits the convergence detection problem. The

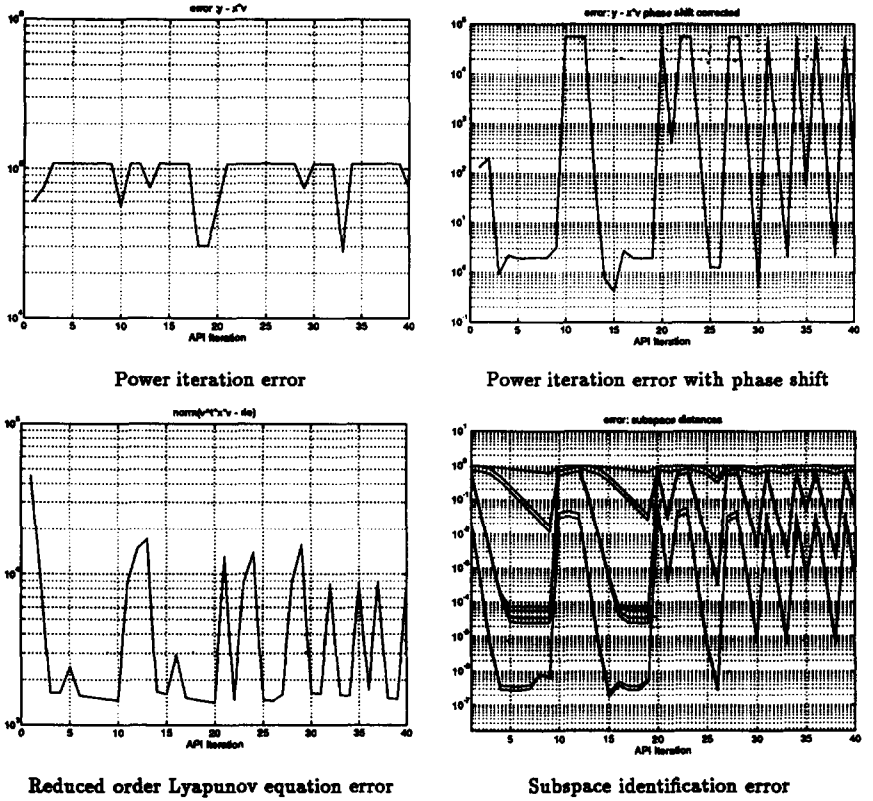


FIG. 9. API results: Jordan-form ACES example.

least-squares form of the algorithm performs significantly worse than the previous version, as is consistent with our expectations.

5. CONCLUSIONS

In this paper we have presented the approximate power iteration (API) algorithm for the identification of the dominant rank-k subspace of the solution X of the Lyapunov equation. The algorithm is heuristically motivated by (1) Krylov-subspace techniques for iterative solution of the Lyapunov equation and (2) power iteration methods for low-rank approximation of large symmetric positive definite matrices. The API algorithm computes estimates of matrix-vector products involving the (implicitly specified) solu-

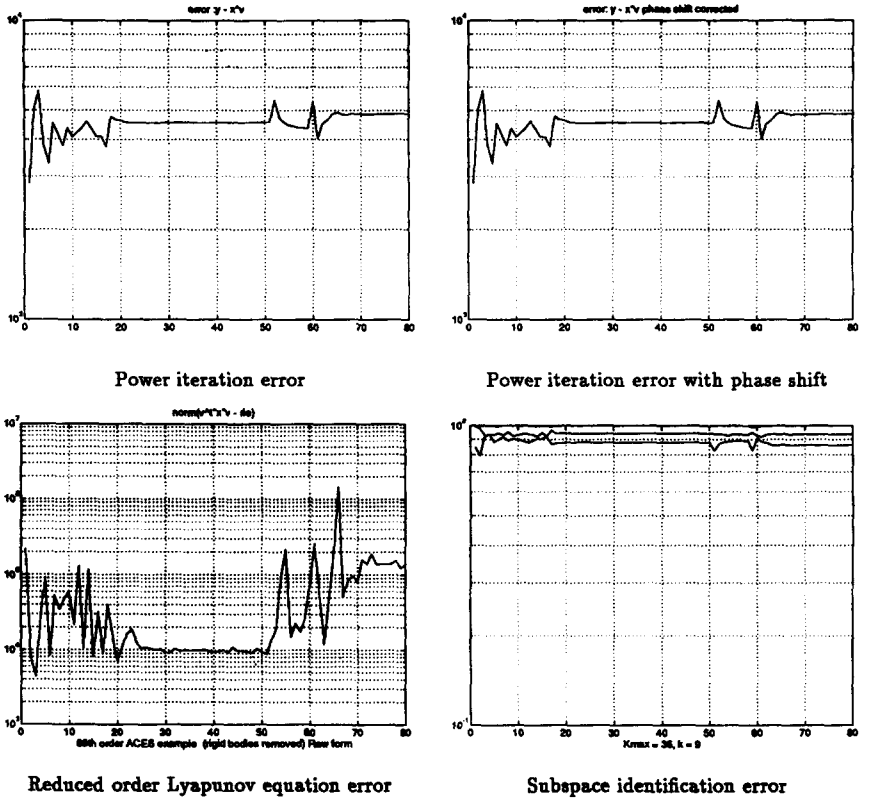


FIG. 10. API error: ACES example, least-squares approximation method.

tion X of the Lyapunov equation (1.1). Our theoretical analysis shows that, under conservative conditions, a naive rank-1 API algorithm converges to a small attractive region in $\mathbb{R}^n$ containing a fixed point v_1 that is near the dominant eigenvector u_1 of X; this analysis requires that the coefficient matrix A of the Lyapunov equation satisfy $A + A^T < 0$. The theoretical result is bolstered by numerical examples in which the API algorithm accurately identifies the rank-k dominant eigenspace of the Lyapunov-equation solution X, for systems satisfying $A + A^T < 0$. Two of the five examples presented satisfy the above negative-definiteness condition $A + A^T < 0$; however, the algorithm fails to identify any of the dominant eigenvectors of the three examples which do not satisfy this restriction. A naive implementation of a preconditioning procedure is implemented on these examples to satisfy the above condition, with limited results. Development of effective

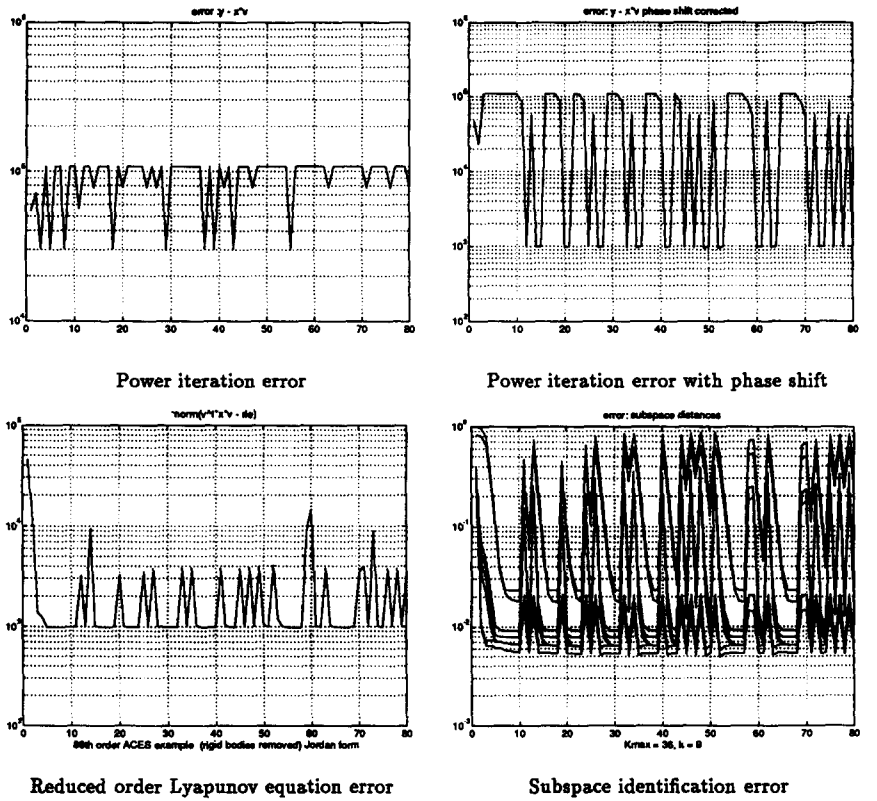


FIG. 11. API error: ACES example, Jordan form, least-squares approximation method.

preconditioning techniques for systems whose A matrices have sign-indefinite symmetric parts remains an open problem. We conclude that, on model problems, API is a highly effective algorithm for approximation of the dominant invariant subspace of the solution X of the Lyapunov equation.

REFERENCES

1 Vasile I. Istrăţescu, *Fixed Point Theory*, Reidel, Dordrecht, 1981.
2 B. D. O. Anderson and Y. Liu, Controller reduction: Concepts and approaches, *IEEE Trans. Automat. Control* 34(8):802–812 (1989).
3 R. H. Bartels and G. W. Stewart, Solution of the matrix equation $AX + XB = C$, *Comm. ACM* 15(9):820–826 (Sept. 1972).

4 Emmanuel C. Collins Jr., Douglas J. Phillips, and David C. Hyland, Robust decentralized control laws for the ACES structure, *IEEE Control Systems* 11(4):62–70 (Apr. 1991).

5 Nancy E. Elsner and Eugene L. Wachspress, Alternating direction implicit iteration for systems with complex spectra, *SIAM J. Numer. Anal.* 28(3):859–870 (June 1991).

6 G. H. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins U.P., Baltimore, 1989.

7 S. J. Hammarling, Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.* 2(3):303–323 (July 1982).

8 G. Hewer and C. Kenney, The sensitivity of the stable Lyapunov equation, *SIAM J. Control Optim.* 26(2):321–344 (Mar. 1988).

9 N. J. Higham, Perturbation Theory and Backward Error for $AX - XB = C$, Numerical Analysis Report 211, Dept. of Mathematics, Univ. of Manchester/UMIST, 1992.

10 A. Scottedward Hotel, Numerical Methods for the Solution of Large and Very Large, Sparse Lyapunov Equations, Ph.D. Thesis, Univ. of Illinois at Urbana-Champaign, 1989.

11 A. Scottedward Hodel, Least squares solution of overdetermined Sylvester equations, in *Proceedings of the 30th Allerton Conference on Communications, Control, and Computing*, Monticello, Ill., 1993, to appear.

12 A. Scottedward Hodel and Kameshwar Poolla, Heuristic methods to the solution of very large, sparse Lyapunov and algebraic Riccati equations, in *Proceedings of the 27th IEEE Conference Decision and Control*, Austin, Tex., 7–9 Dec. 1988, pp. 2217–2222.

13 A. Scottedward Hodel and Kameshwar R. Poolla, Parallel solution of large Lyapunov equations, *SIAM J. Matrix Anal. Appl.* 13(4):1189–1203 (Oct. 1992).

14 D. Y. Hu and L. Reichel, Krylov subspace methods for the Sylvester equation, *Linear Algebra Appl.* 172:283–313 (1992).

15 Imad M. Jaimoukha and Ebrahim M. Kasenally, Krylov subspace methods for solving large Lyapunov equations, *SIAM J. Numerical Analysis*, 1994, volume 31, pp. 227–251, Feb. 1994.

16 V. L. Jones, Overview of LSS Ground Test Verification Facility at NASA/MSFC and two LSS Models, Technical Report, Control Dynamics Co., Huntsville, Ala., June 1986.

17 P. Lancaster, Explicit solutions of linear matrix equations, *SIAM Rev.* 12(4):544–566 (Oct. 1970).

18 N. Levenberg and L. Reichel, A generalized ADI iterative method, preprint.

19 An Lu and E. L. Wachspress, Solution of Lyapunov equations by alternating direction implicit iteration, *Comput. Math. Appl.* 21:43–58 (1991).

20 B. C. Moore, Principal component analysis in linear systems: Controllability, observability and model reduction, *IEEE Trans. Automat. Control* AC-26(1):17–32 (Feb. 1981).

21 D. P. O'Leary and G. W. Stewart, Data-Flow Algorithms for Parallel Matrix Computations, Technical Report 1386, Univ. of Maryland, Jan. 1984.

22 B. N. Parlett, The software scene in the extraction of eigenvalues from sparse matrices, *SIAM J. Sci. Statist. Comput.* 5(3):590–604 (Sept. 1984).

23 Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 7(3):856–869 (July 1986).

24 Youcef Saad, Numerical solution of large Lyapunov equations, In *Signal Processing, Scattering and Operator Theory, and Numerical Methods: Proceedings of the International Symposium MTNS-89*, (M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, Eds.), Vol. III, Birkhäuser, Boston, 1990, pp. 401–410.

25 M. G. Safonov and R. Y. Chiang, A Schur method for balanced-truncation model reduction, *IEEE Trans. Automat. Control* 34(7):729–733 (July 1989).

26 G. Starke and W. Niethammer, SOR for $AX - XB = C$, *Linear Algebra Appl.* 154–156 (1991).

27 E. L. Wachspress, Iterative solution of the Lyapunov matrix equation, *Appl. Math. Lett.* 1(1):87–90 (1989).

Received 25 February 1993; final manuscript accepted 20 June 1994